D5.9

Design of the Data Mapper and Converter to FHIR - v1

ABSTRACT

This document describes the fundamental software components - together referred to as "data integration platform" - that are responsible for the conversion and translation of Electronic Health Records across languages and across local and national healthcare standards.

Delivery Date	9th October 2019
Work Package	WP5
Task	T5.3
Dissemination Level	Public
Type of Deliverable	Report
Lead partner	UNITN

InteropEHRate project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106.

This document has been produced in the context of the InteropEHRate Project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106. All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

This work by Parties of the InteropEHRate Consortium is licensed under a Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/).

CONTRIBUTORS

	Name	Partner
Contributors	Gábor Bella	UNITN
Contributors	Subhashis Das	UNITN
Contributors	Simone Bocca	UNITN
Contributors	Paolo Bouquet	UNITN
Contributors	Mattia Fumagalli	UNITN
Contributors	Stefano Dalmiani	FTGM
Contributors	Alessio Graziani	ENG
Contributors	Francesco Torelli	ENG
Contributors	Marcel Klötgen	FRAU
Reviewers	François Sevrin	A7
Reviewers	Alessio Graziani	ENG
Reviewers	Juan Fernandez	EFN

LOGTABLE

Version	Date	Change	Author	Partner
0.1	2019-08-31	First version created	Gábor Bella	UNITN
0.2	2019-09-06	Contributed information on medical knowledge encoding	Subhashis Das	UNITN
0.3	2019-09-09	Contributed information on knowledge management tools	Paolo Bouquet	UNITN
0.4	2019-09-09	Contributed information on batch knowledge importing and APIs	Simone Bocca	UNITN
0.5	2019-09-09	Contributed information on domain knowledge development	Mattia Fumagalli	UNITN
0.6	2019-09-13	Finalised version ready for review	Gábor Bella	UNITN
0.7	2019-09-13	Review with comments	Juan Fernandez	EFN

0.8	2019-09-17	Review with comments	François Sevrin	A7
1.0	2019-10-02	Quality check	Argyro Mavrogiorgou	UPRC
1.1	2019-10-04	Final review	Laura Pucci	ENG
1.2	2019-10-08	Addressed additional comments	Gabor Bella	UNITN
Vfinal	2019-10-09	Final version for submission	Laura Pucci	ENG

ACRONYMS

Acronym	Term and definition
API	Application Programming Interface
ConversionI	Application Programming Interface of the InteropEHRate Conversion Services component, for the conversion of content and structure within EHRs into a SEHR compliant form.
CSV/TSV	Comma-Separated Values / Tab-Separated Values: simple machine-readable tabular file formats.
EHR	Electronic Health Record (e.g., as provided by a hospital).
IHS	InteropEHRate Health Services: a high-level software component (a collection of libraries) that provides high-level EHR translation and conversion services to end-user applications.
IHSI	Application Programming Interface of the InteropEHRate Health Services component that provides high-level EHR transformation services.
IHT	InteropEHRate Health Tools: a set of interactive helper tools that are used by hospital employees (data scientists) to set up and maintain the EHR data integration system. These tools are outside the scope of the project deliverables and thus are not fully specified in any deliverable document.
JSON	JavaScript Object Notation: standard, computer-readable, tree-structured file format.
LEI	Legacy EHR Interface: Application Programming Interface implemented by local health institutions (hospitals) to provide a unified access to EHRs to InteropEHRate services.
SEHR	Smart Electronic Health Record: the interoperable, multilingual, standard, FHIR-based representation of EHRs as defined and used by the InteropEHRate project.
TranslationI	Application Programming Interface of the InteropEHRate Translation Services component, for the translation of natural language text within EHRs.
XML	Extensible Markup Language: machine-readable tree-structured file format.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Scope of the document

This deliverable provides a first description of the services, software components, and processes responsible for converting local hospital EHRs into the interoperable SEHR form that is proposed by the InteropEHRate project.

1.2. Intended audience

This document adopts a "service" perspective in the sense that it concentrates on how data conversion services can be called and used by applications (such as the *HCP App*), and also on the mechanisms by which these services are set up and maintained by hospitals. As opposed to deliverable D5.7 (*Health Data Integration Platform*) that defines the low-level operations and inner workings of data integration, this deliverable defines its high-level interfaces, services, and usage processes.

1.3. Structure of the document

Section 2 provides the high-level architecture of the InteropEHRate Health Services, and within them the Conversion Services and the Health Data Integration Platform. Section 3 gives an overview of the general principles of the knowledge-driven approach to EHR integration and conversion. Section 4 presents in detail the process by which health knowledge and the EHR integration pipeline that uses it need to be set up, used, and maintained. Section 5 presents high-level information with respect to APIs and implementation of the software components involved in EHR conversion. Section 6 provides details about the international (cross-border) knowledge that is being set up for the project following the guidelines of section 3. Section 7 gives conclusions and addresses future work.

1.4. Updates with respect to previous version (if any)

Not applicable.

2. INTEROPEHRATE HEALTH SERVICES ARCHITECTURE

InteropEHRate Health Services (IHS) is a set of high-level software services provided to hospitals and health applications in order to carry out operations related to health record interoperability. Depicted in the centre of the diagram below, it contains the following components:

- a central *Data Integration Platform*, as specified in [D5.7];
- two high-level interoperability service implementations:
    - *Conversion Services:* specified in this deliverable, their role is to convert EHRs from the format locally used by a hospital to a cross-border interoperable format;
    - *Translation Services:* specified in deliverable [D5.11], their role is to translate and localise natural-language text contained in EHRs from the original source language to other European languages, such as the language of the patient or the language locally used by the hospital, as required by interoperability;
- a *Controller* component, specified in this deliverable, that serves as middleware between the interoperability services, the Hospital Information System, and applications.

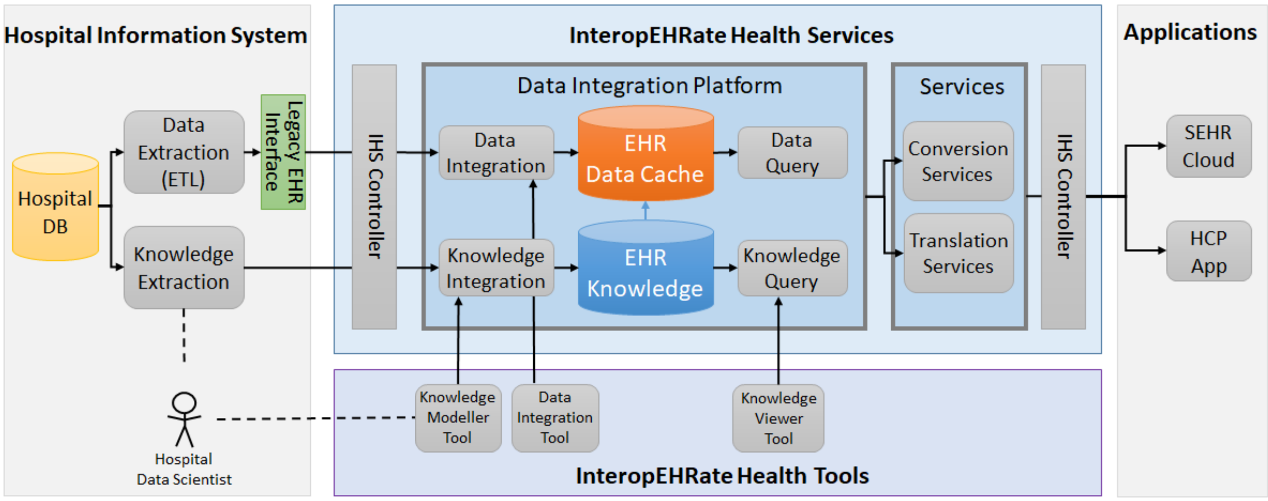

Figure 1 - The architecture of InteropEHRate Health Services and its relation to the Hospital Information System, Applications, and InteropEHRate Health Tools

3. PRINCIPLES OF KNOWLEDGE-DRIVEN CONVERSION

The particularity of the solution to data conversion adopted in the project, originally conceived by UNITN, is that it is entirely knowledge-driven: instead of being hardcoded into business logic, data conversion rules and methods are rather encoded dynamically in a flexible manner. This has the advantage of making the adaptation and the evolution of conversions a much more agile process: instead of going through costly software development cycles, the system can be adapted to changes in data structures, encoding standards, etc., virtually by pointing and clicking using interactive tools, the changes being immediately effective on the whole system.

The price of this flexibility is the need formally to encode all knowledge necessary for the conversions. This includes locally relevant knowledge (locally used terminologies, data schemas, etc.), international knowledge as used by InteropEHRate, as well as mappings between the two. As international knowledge is, in principle, identical for every participating country and institution, it will be pre-loaded into all deployable instances of the IHS. All local institutions need to do is formalise their locally relevant knowledge (only to the extent necessary for the conversions) and define their mappings to international knowledge. This is what is meant by *knowledge extraction* and *knowledge integration* in fig. 1 above (see the next section for details).

The table above shows the different kinds of knowledge that need to be integrated into the *EHR Knowledge Base* (see fig. 1) for conversions to be fully operational. The *EHR Knowledge Base* encodes knowledge in a layered architecture where each layer is constructed from elements from the ones below it:

- **the lexicon layer** encodes the meaningful and relevant words, domain terms, and codes that are used in local EHRs, in the languages supported by the IHS system;
- **the concept layer** encodes the supra-lingual meanings of words, terms, and codes; it is organised into a hierarchical graph and acts as a bridge across languages;
- **the schema layer** encodes data structures used to describe types of entities relevant to EHRs, such as patients, hospital visits, prescriptions, medicine, laboratory tests, etc.;
- **the entity layer** encodes data instances that, contrary to concepts, are richly described, such as substances or drug products.

The table below gives a summary of the kinds of knowledge contained in the EHR Knowledge Base.

	Local	International	Mappings
entities	optional: richly described locally used data instances (e.g., drug substances described with multiple attributes)	internationally defined data instances (e.g., drug substances richly described by ATC)	equivalence of local and international entities
schemas	optional: locally used data structures	international data structures, i.e., FHIR	structural mappings and data transformations (defined in the Data Pipeline Setup step)
concepts	the meanings of locally used terms, encoded as nodes of a graph	international encoding systems, e.g., ICD or LOINC	equivalences or is-a relations between local and international terms

lexicon	natural language labels used for local concepts, including local codes	labels of international codes	equivalences between labels expressing the same meaning, both within and across languages

Table 1- Kinds of knowledge

4. EHR CONVERSION PROCESSES AND ROLES

In order for EHR conversion to be an automatic process, the Platform needs to be set up and customised with respect to the data representations used in the hospital. This section describes the entire process by which conversion is set up, then used on a daily basis, and also regularly maintained to follow the natural evolution of data representations, both local and international. This process is divided into four macro-steps:

1. **[ONLY ONCE] Knowledge extraction and integration:** local and international coding systems, terminologies, data schemas, and mappings are semi-automatically encoded as formal knowledge.
2. **[ONLY ONCE] Data pipeline setup:** data structure and data value transformations are formally defined in a semi-automatic way, in the form of a *conversion recipe*.
3. **[DAILY] Automated data extraction and integration:** the conversion recipe is automatically applied to EHRs that need to be transformed to the international format.
4. **[YEARLY OR AS NEEDED] Knowledge and data pipeline maintenance for data evolution:** elements of the knowledge and/or the recipe are modified so that they keep up with the evolution of data representations and encoding standards.

Two roles are involved in carrying out the steps above, typically both provided by the local healthcare provider institution (hospital).

- **A data scientist** is the primary person that executes the steps above. (S)he has a good background in healthcare data and knowledge modelling as well as familiarity with the local data modelling practices. Typically (s)he has some programming knowledge but does not need to be a bona fide software developer.
- **A software developer** has a supporting role behind the data scientist, with the goal of automating repetitive modelling tasks. For example, the integration of a coding system with tens of thousands of codes is best executed as a batch task that may require procedural automation (i.e., through scripting). Such one-shot tasks are best carried out by a software developer.

4.1. Knowledge Extraction and Integration

Knowledge extraction and integration are once-and-for-all tasks that are executed in an initial bootstrapping phase of the deployment of the IHS at a local healthcare institution. As international knowledge is pre-loaded into the knowledge base, this macro-operation concerns only local knowledge and its mapping to international knowledge. The operation is divided into the following steps:

1. **Local knowledge extraction:** this is a fully manual analysis of local EHR datasets in order to interpret their contents in terms of the four formal knowledge layers. The result of this activity is a document that defines the subset of elements (strings, labels, codes) in local data that will need to be represented as formal knowledge.
2. **Local knowledge and mapping integration:** in this step, the data scientist and/or the software developer takes the document above as input and, using the IHT and scripting, produce formal knowledge ready to be imported into the knowledge base. Knowledge integrated at this point does not need to describe local data exhaustively: it is enough to define the mappings that are necessary to convert locally used terms, codes, and data structures into international knowledge. Such mappings are defined in one of the following ways:
    - **on the lexical and concept levels:**
        - if the concept (meaning) of a locally used term has an equivalent concept in the international knowledge (such as a local ICD-9 code “433.00” being equivalent to

    the international ICD-10 code “i65.1”), define the label of the local term to be a synonym of the international term,
    - if the concept (meaning) of a locally used term does not have an equivalent concept in the international knowledge, create the new concept as broader or narrower than the closest existing international concept (if any), and add the local term as its lexicalisation;
    - the steps above are executed:
        - through a GUI when the number of natural language terms and their meanings (concepts) to add is low (up to a few dozen),
        - through filling in an Excel sheet manually for a low to medium number of terms and/or concepts (up to a few hundred),
        - through batch importing of automatically generated Excel sheets, in case the number of terms and their concepts is high (thousands to millions);
- **on the schema level:** the formal definition of local schemas is optional; however, it can be useful for the purposes of locally harmonising heterogeneous data and in order better to define its meaning; it can be done:
    - through a GUI when the number of schemas and attributes to define is low (less than a dozen of schemas, less than about a hundred attributes);
    - automated batch importing of OWL schema definitions, for larger numbers of schemas and attributes.

The screenshots below show the GUI of the Knowledge Modeller Tools from IHT.

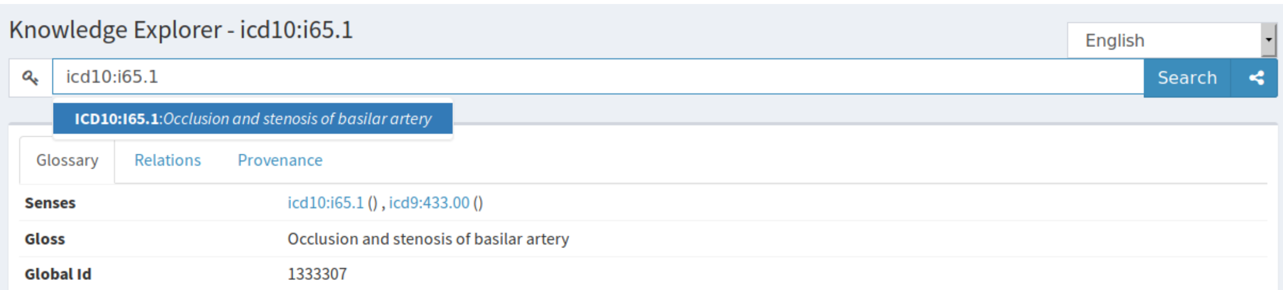

Figure 2 - Screenshot from the concept and terminology explorer tool, showing a coded value from ICD-10, its meaning, its mapping to ICD-9, and its internal concept ID.

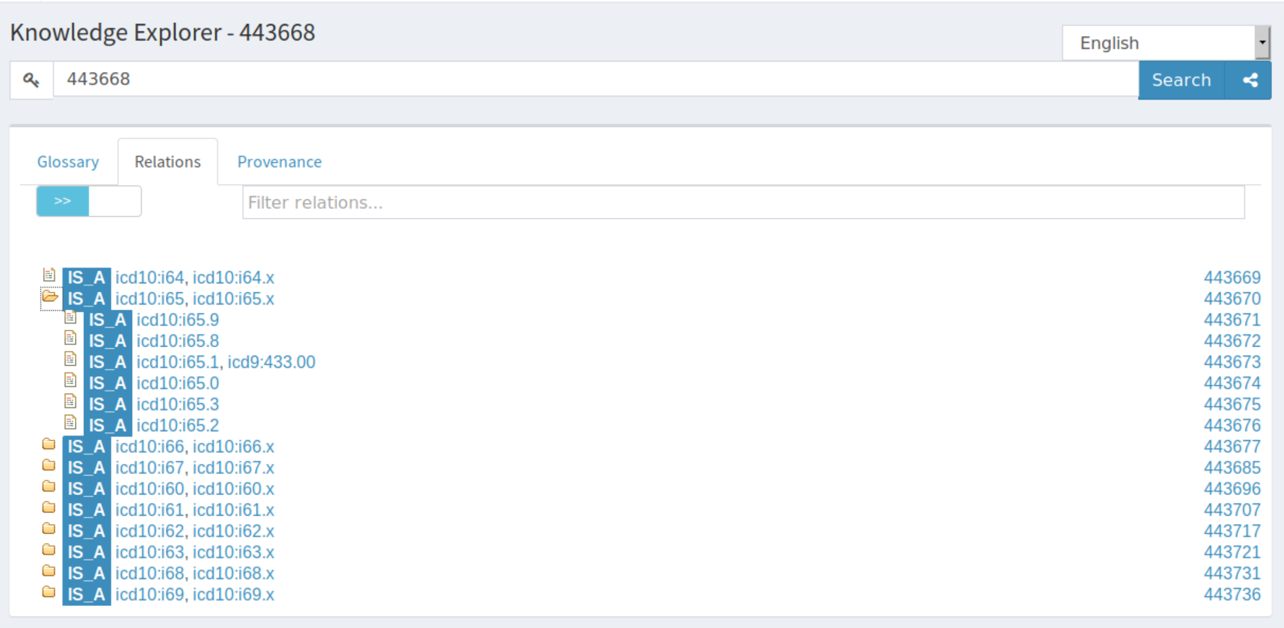

Figure 3 - Screenshot from the concept and terminology explorer tool, showing the same ICD concept (i65.1) within the ICD hierarchy.

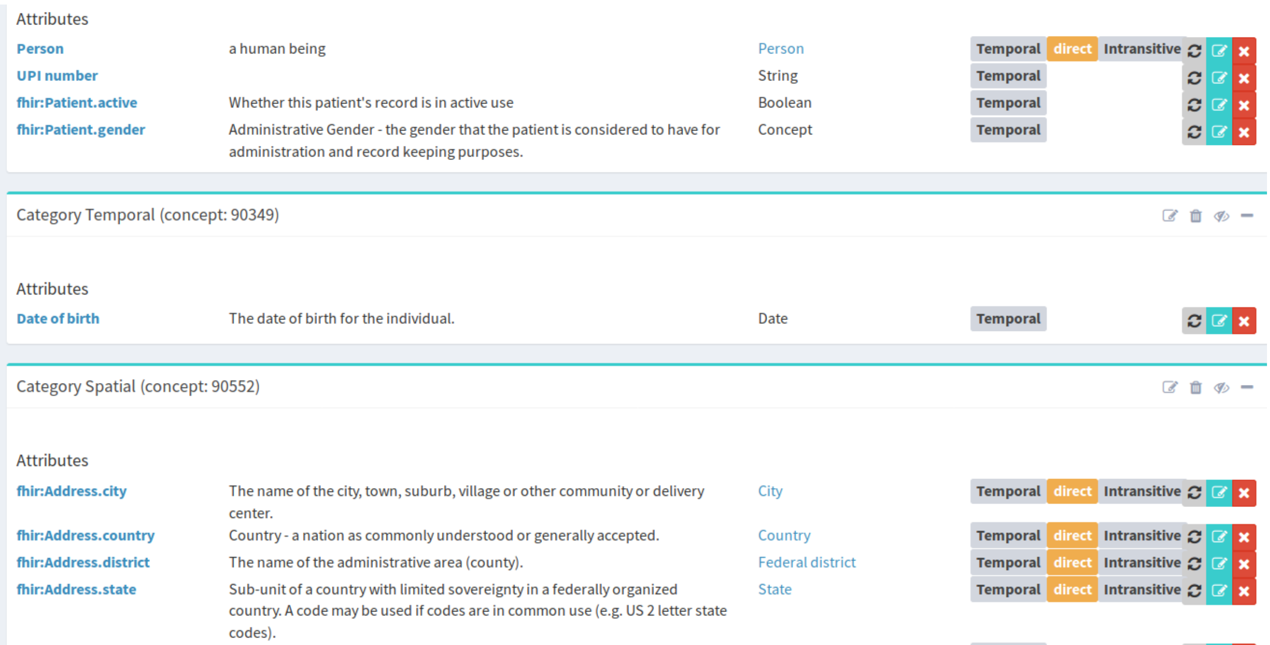

Figure 4 - Figure 4: Screenshot from the entity type modeller tool.

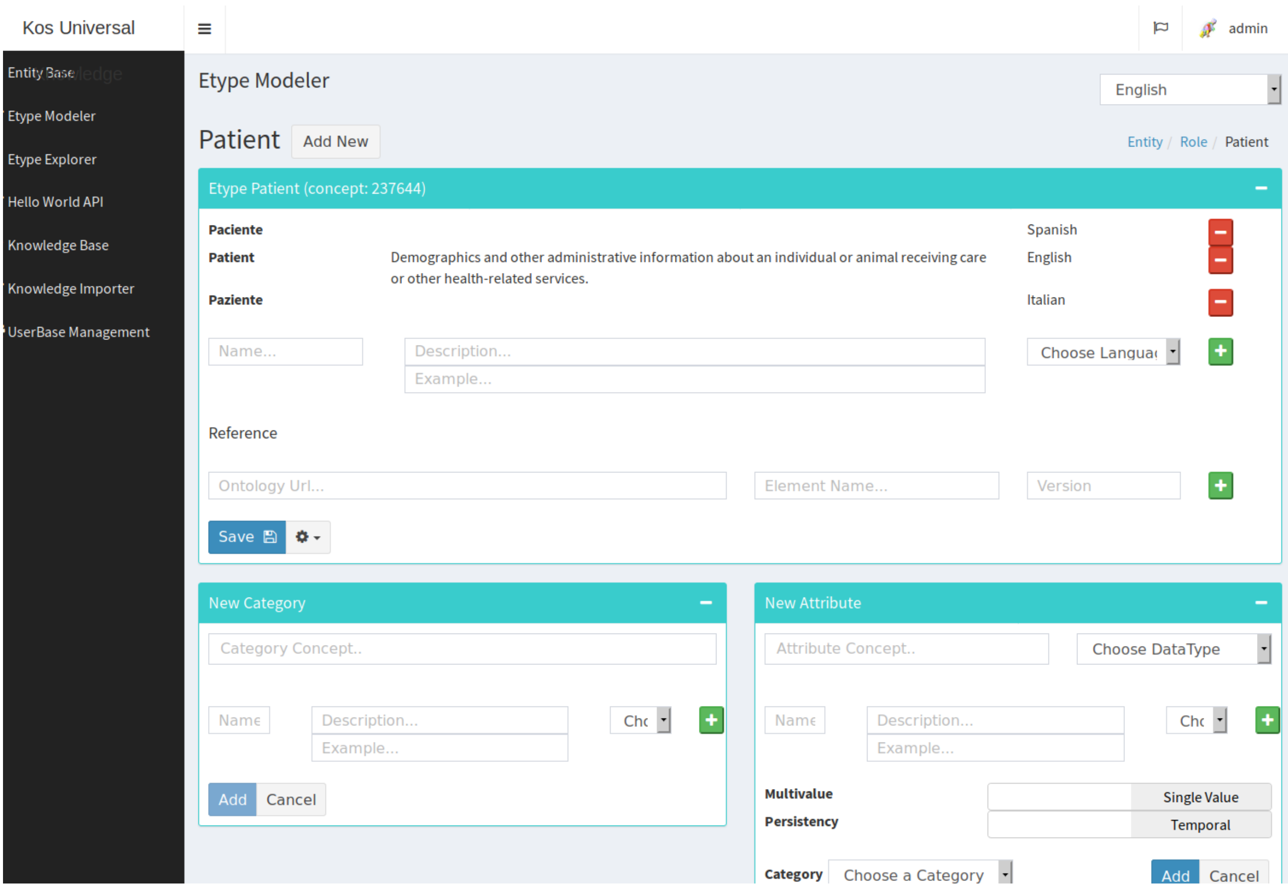

Figure 5 - Screenshot from the entity type modeller tool.

4.2. Data Pipeline Setup

After setting up knowledge mappings, in this step the data integration, i.e., data transformations from local EHR formats to the FHIR-based SEHR format, is defined. This process only needs to be done once, in the bootstrapping phase of the IHS. It is done through two steps:

1. **data extraction setup:** simple surface-level data format conversion in order to provide data to the data integration tool in a format that it accepts;
2. **data integration setup:** this is the step that performs the structural conversion to FHIR and also the meaning-level conversions of data values.

4.2.1. Data Extraction Setup

Data extraction consists typically of a simple script that converts "raw" EHR data (i.e., as extracted from local DBs) into the formats that are accepted by the following data integration step. This adaptation step is also necessary for reasons of data privacy and security: the data extraction code, typically implemented by the local institution, acts as a separation layer between local logic and the systems deployed as part of IEHR.

The input of data integration, which is the output of data extraction, must respect the following requirements:

Category	Requirement
Format	CSV, XML, or JSON, always well-formed
Character encoding	UTF-8 is mandatory
Structure	tabular (preferred) or tree-structured (acceptable)
Level of aggregation (number of attributes per file)	It is typically easier for the data scientist to work with multiple input files with a smaller number of attributes (10-20) rather than with one file containing hundreds of attributes.
Depth of embedding	In the case of tree-structured data, for a better understanding and usability by the data scientist in the data integration step, the levels of embedding should not be too deep (typically not more than 3-4 levels) and the tree should not be too complex. In the case of a wide and/or deep and/or complex dataset, splitting it into smaller parts enhances usability.
Number of records	Data integration can deal both with a single record (EHR) and with multiple records in the same file.
Noisiness	While it is possible to deal with noise in the successive data integration step, the more noise is eliminated upstream, the less work the data scientist needs to do.
Metadata	Metadata accompanying the data (e.g., provenance, timestamps) must be represented following the DCAT standard.

Table 2 - Input Requirements

4.2.2. Data Integration Setup

Data integration is set up through a graphical tool called StarLinker that allows the data scientist to map local data schemas to the FHIR standard. The syntactic (mandatory) mappings define:

- which local schema(s) are mapped to which FHIR resource(s);
- which local attribute(s) are mapped to which FHIR attribute(s);
- how the data contained in each attribute needs to be transformed to be FHIR-compliant;
- additional data cleaning operations, if necessary.

In addition, the semantic (recommended) mappings define:

- the language of potentially multilingual attribute values;
- how health terms, including healthcare codes, within data values are converted into concepts.

Once all mappings are defined, the data scientist saves the entire mapping process as a *data integration model*. This model serves as a recipe for the subsequent automation of EHR conversions.

The figure below shows a screen capture from the interactive tool used for defining the mappings.

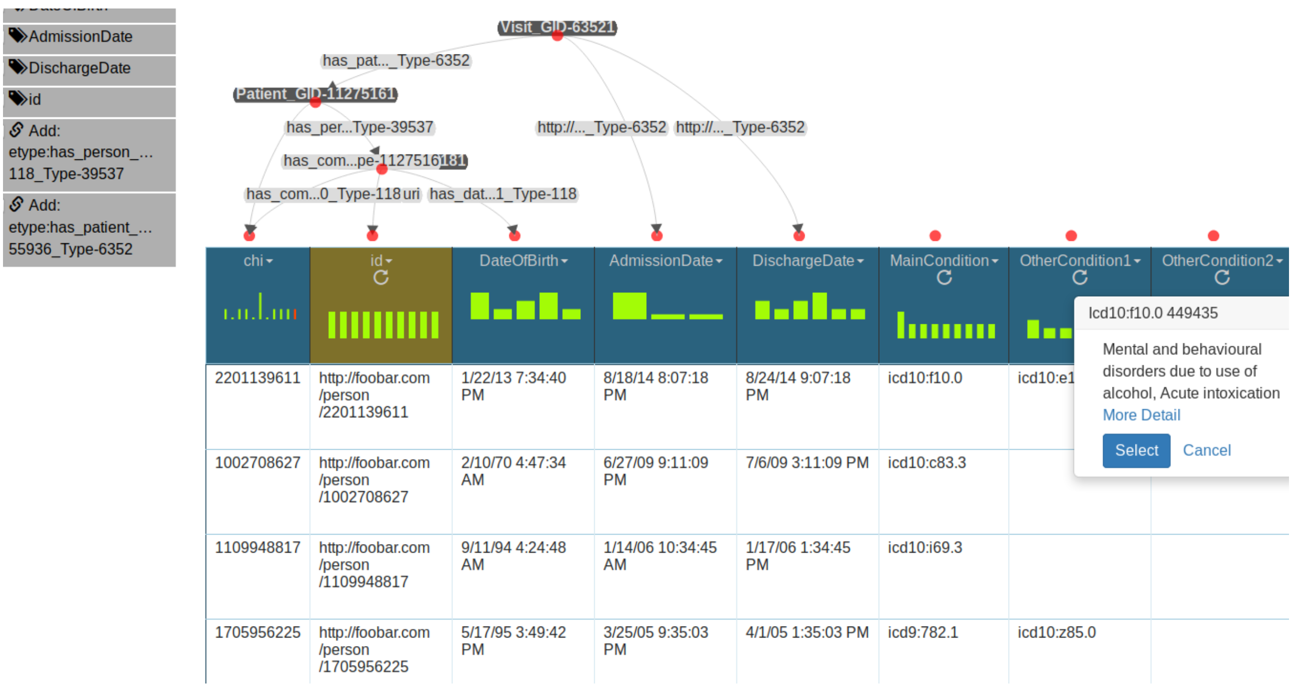

Figure 6 - The StarLinker tool UI for performing data integration setup; part of the "recipe" is visible in the upper left corner

4.3. Automated Data Extraction and Integration

Once both the extraction script and the *data integration model* for EHR conversion have been generated in the previous step, they can be reapplied to any number of EHRs in a fully automated manner. This is achieved using the StarLinker tool in batch (non-interactive) mode.

Fully automated conversion is the "usual" modus operandi of the IHS. While the steps above are executed once and for all in a bootstrapping phase of the system, automated data extraction and integration is called for each SEHR request coming from a client, such as the HCP app.

All automated methods of the data conversion APIs are described in section 5 below. As defined in [D2.4], data extraction and integration are always executed at the source institution. This is because only the originating institution has sufficient knowledge of its own data representations to be able to define mappings precisely and to keep them up to date with the constant evolution of local data.

While not explicitly within the scope of the InteropEHRate project, the Conversion Services also provide methods for data conversions at target institutions that need to deal with international SEHRs brought in by patients. While the InteropEHRate architecture requires target institutions to be able to interpret SEHRs directly (that is, to have support for FHIR and ideally also for international terminologies as used in SEHRs), the Conversion Services provide methods for mapping international terms to local ones, probably easier to interpret for local HCPs. This functionality obviously relies on the terminological mappings to be formally defined as part of the local knowledge.

In case the input EHR format (structure and/or content) changes in a major way that is not compatible anymore either with the extraction script or the data integration model generated (e.g., the attribute names are modified), then the automated conversion may produce unexpected results or fail entirely. In such cases a pipeline maintenance step (see below) is necessary.

4.4. Knowledge and Data Pipeline Maintenance for Data Evolution

The recipe-based automation of data integration is robust to a certain extent: it is able to deal with previously unseen data values (such as codes, text, or attributes) provided that such values are already encoded as formal knowledge. In case they are not encoded and the change is monotonic (the existing knowledge remains intact), the automated integration process will not fail: it will simply ignore data that it cannot recognise. In the case of non-monotonic changes, an explicit failure may occur.

The strength of the IEHR data integration process is that the system is able to follow most of the typical cases of data evolution by simple, agile changes to the underlying knowledge, as opposed to having to modify the source code through software development cycles. The table below lists the possible knowledge and/or data maintenance operations that may occur due to data evolution.

Kind of data evolution	Example	Maintenance operation	Executed by
New health term or coded value	A new health intervention code	Adding the new term as a new word, and also as a new concept if necessary, into the knowledge through the IHT Knowledge Management Tool.	Local data scientist
New or evolving encoding standard	Adoption of ICD-11	Adding the codes of the new standard as new words (and new concepts for previously non-existent meanings) into the knowledge through batch knowledge import.	Local software developer assisted by local data scientist
New input data attribute	Addition of a new element in the history of the patient	Manual extension of the existing *data integration model* by mapping the new attribute using StarLinker	Local data scientist
Evolving international data schema	evolution of FHIR v4 to FHIR v5	Extension of the FHIR pivot schemas (entity types) in the formal knowledge and update of the data integration model (attribute mappings) using StarLinker	IEHR (non-local) data scientist
Major, disruptive change in the input format	The hospital DB schemas are completely reorganised	Rewriting of the data extraction script and manual generation of a new data integration model using StarLinker	Local data scientist, local SW developer
Major, disruptive change in the international data schema	FHIR is replaced by another standard	Creation of new entity types in the formal knowledge representing the new international data schema, followed by the update of the data integration model using StarLinker	IEHR (non-local) data scientist, local data scientist

Table 3 - Data Evolution and operations

5. INTERFACES AND IMPLEMENTATION OF DATA CONVERSION SERVICES

This section describes both the interfaces for accessing the conversion services, and the interfaces through which these services connect to the underlying Platform. Calls from external applications go through a series of API layers, from the most project-specific to the most generic.

1. External IEHR applications, such as the HCP App, should call the *IHS Interface* (IHSI) for high-level services that are based on standards and solutions adopted by the InteropEHRate project.
2. The *IHS Controller* takes such high-level calls and decomposes them into lower level *EHR retrieval, conversion,* and *translation* calls, through the *LEI (Legacy EHR), ConversionI,* and *TranslationI* interfaces, respectively. These three interfaces are more generic than IHSI in the sense that they are agnostic of the standards stack of InteropEHRate.
3. Finally, the *EHR Conversion Services* implementation decomposes requests to *ConversionI* to low-level requests to the *Data Integration Platform*. The Platform is even more generic as it is agnostic with respect to the EHR interoperability use case.

The figure below provides the component diagram for the interfaces and components within the IHS.

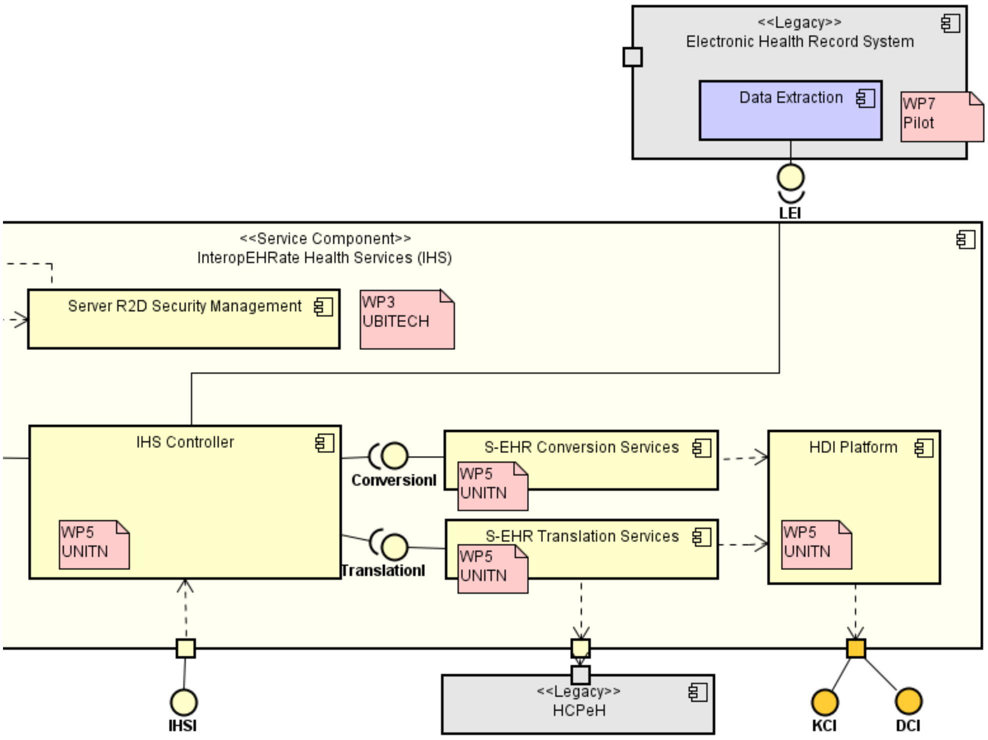

Figure 7 - Relevant detail from the IHS component diagram showing the interfaces and components discussed in this section

Below we present the IHSI and ConversionI interfaces. The Platform interface is presented in the Platform deliverable [D5.7].

5.1. The InteropEHRate Health Services Interface (IHSI)

The IHS Controller (see figure 1) acts as a middleware between calls from external applications (such as the HCP App) and SEHR conversion and translation functionalities. Its role is to adapt the latter operations to the specific requirements of the local hospital context: the local language, the local standards used, the local IT system, etc. Therefore, through the IHSI it receives high-level calls that are agnostic of local specificities. The table below shows the IHSI endpoints:

IHSI Endpoint	Input	Output	Description
retrieveSEHR	String patientID, LEVEL level, Lang patientLang	SEHR sehr	retrieves the EHR specified through its local ID from the local DB system, converted into SEHR on the interoperability level LEVEL (SEC, SYN, or SEM), and optionally translated into the patient's language given as input
localizeSEHR	SEHR sehr	SEHR localizedSEHR	transforms a SEHR in input into a form suitable to be used locally, i.e., translated into the local language and using local standards
persistSEHR	SEHR sehr	PersistResult result	takes a request for persisting a SEHR into the local DB system and delegates it to the local IT infrastructure
localizeTerms	String[] srcTerms, String srcStd	String[] trgTerms	converts a list of source terms (e.g., codes) from a source standard to its local equivalents, if they exist

Table 4 - IHSI endpoints

5.2. The Legacy EHR Interface (LEI)

This interface, implemented by local institutions, provides access to the local (legacy) EHR information system in a controlled way. It provides the following endpoints:

LEI Endpoint	Input	Output	Description
retrieveEHR	String ehrID	File[] files	retrieves the EHRs specified through their local IDs from the local DB system and returns them as a set of files
persistSEHR	SEHR sehr	PersistResult result	persists a SEHR into the local information system

Table 5 - LEI endpoints

5.3. The Conversion Interface (ConversionI)

The *EHR Conversion Services* are a set of medium-level services provided through the *ConversionI* API. The implementation of these services has a double role:

- it translates high-level requests for converting entire EHRs into a series of lower-level requests to the Data Integration Platform;
- it acts as an abstraction layer on top of the Platform, hiding its specific implementation details.

The API of the Conversion Services provides the following endpoints, called from the IHS controller but also callable by any third-party application needing to use the knowledge-based data conversion services:

ConversionI Endpoint	Input	Output	Description
convertEHR	File[] inputFiles, DataIntModel model, SEHRFormat outputFormat	File[] outputFiles	converts a local EHR, represented as a list of CSV/XML/JSON files, into an interoperable format as defined by the input data integration model, and returns it in a serialised format outputFormat
retrieveConvertedEHR	String ehrID, SEHRFormat outputFormat	File[] outputFiles	retrieves the already converted EHR specified through its local patient ID from the EHR Data Cache, and returns it in a serialised format outputFormat
convertTerms	String[] srcTerms, String srcStd, String trgStd	String[] trgTerms	converts a list of source terms (e.g., codes) from a source standard to its equivalents, if they exist, in the target standard

Table 6 - ConversionI endpoints

5.4. EHR Conversion Sequence Diagram

The diagram below shows how the interfaces above are called to implement the conversion of a legacy EHR into a SEHR.

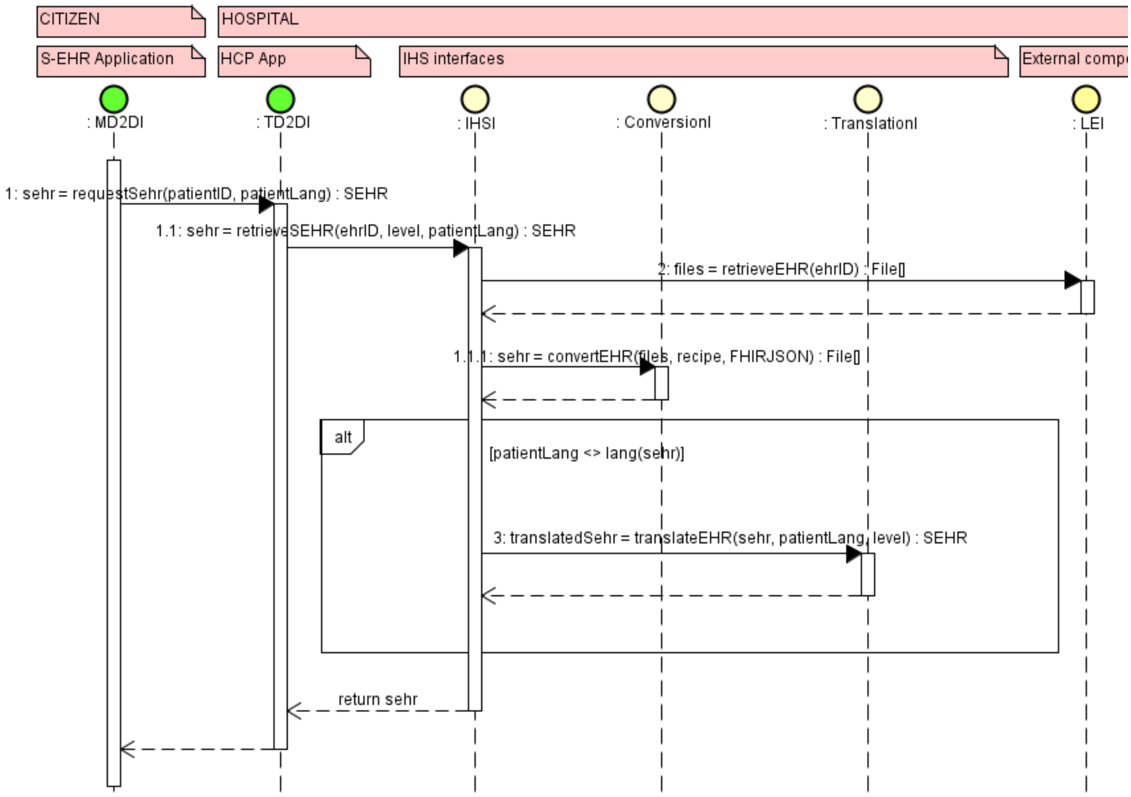

Figure 8 - Sequence diagram of a request, coming from the HCP App, to retrieve an existing EHR and convert it into a SEHR

6. INTERNATIONAL KNOWLEDGE AND ITS MAPPINGS

This section presents the *international knowledge* content developed for specific IHS installations in preparation for the project scenarios. By *international knowledge* we understand 'pivot' terminology, concepts, and data structures used for interoperability within SEHRs. The standards adopted for international knowledge have been defined in deliverable [D2.7]. The final version of the current deliverable (v2, D5.10) will describe the implementation of the following standards:

- data structures:
    - FHIR v4.0 profile for International Patient Summary,
    - other FHIR profiles as required by the three project scenarios;
- concepts:
    - FHIR concepts,
    - ICD-9 and ICD-10 concepts,
    - SNOMED CT International concepts,
    - LOINC concepts,
    - general language concepts,
- natural language labels (in English, French, Romanian, Greek, and Italian):
    - FHIR labels,
    - ICD labels,
    - SNOMED CT labels,
    - LOINC labels,
    - general language labels;
- entities (structured instances):
    - ATC pharmaceutical substances (drugs).

InteropEHRate proposes an iterative methodology for developing support for these standards. Such a methodology adapts to the general project management methodology where development has to proceed based on partial information. The formalisation of all international knowledge above, except for the translation of language-specific labels, is being carried out by an InteropEHRate data scientist employed by UNITN, with experience in healthcare informatics. The results will be reviewed by healthcare provider partners from the project. In the case of natural language labels where translations do not exist in any of the four target languages (Romanian, Greek, Italian, French), the help of local domain experts from among the project partners will be necessary.

6.1. FHIR IPS Support

FHIR v4.0 IPS (http://hl7.org/fhir/uv/ips/2019Sep/) is an international standard for representing *patient summaries* in a cross-border interoperable form. It is the FHIR-based version of the *International Patient Summary* as defined by HL7. According to the methodology described in section 4, the FHIR IPS data structures will be represented as three separate levels of knowledge:

- *natural language labels* of the data attribute and resource names in all five supported languages;
- *concepts* representing the meanings of data attributes and entity types (FHIR resources);
- *entity types* (data structures) themselves, represented in a language-independent manner using concepts only.

FHIR IPS concepts (there are over 500 of them) and entity types (there are over 20 of them) will be described manually by a data scientist. The concepts will be mapped to equivalent SNOMED CT concepts whenever such an equivalence exists. The corresponding English labels will be taken from the original FHIR specifications. As FHIR is not available in the remaining four languages, translated labels in these languages will have to be provided (for all concepts or at least a subset of the concepts) by local domain experts.

6.2. ICD Support

The *International Classification of Diseases* (https://www.who.int/classifications/icd/en/) is an international standard used in all project partner countries. While its most recent version is 11, most countries are still using version 10, while Italy is using version 9. For this reason, in the project both ICD-10-CM and ICD-9-CM will be used and formally represented:

- ICD-10-CM will be used to define the concept hierarchy;
- ICD-10-CM codes will be defined as natural language labels in all supported languages, prefixed by the standard identifier (e.g., "icd10:i65.1");
- natural language descriptions (e.g., "Occlusion and stenosis of basilar artery") in all supported languages will be used as definitions ("glosses") for the codes; the translations of ICD-9 or ICD-10 (whichever applicable) will have to be provided to the project by national healthcare provider partners;
- equivalent ICD-9-CM codes will be mapped as synonymous labels to the existing ICD-10-CM concepts (e.g., "icd9:433.00" and "icd10:i65.1" are synonyms).

Definitions of the ICD-9-CM and ICD-10-CM hierarchies were provided by the *Centers for Disease Control and Prevention (https://www.cdc.gov), Department of Health & Human Services, USA*. Mappings between ICD-9 and ICD-10 were provided by the *Centers for Medicare & Medicaid Services (CMS) (https://www.cms.gov)*, part of the *Department of Health and Human Services (HHS), USA*.

6.3. SNOMED CT Support

SNOMED CT (http://www.snomed.org/) is becoming a general-purpose reference terminology for interoperability in the healthcare field. Therefore, the project is adopting SNOMED CT both as one of its "pivot" terminologies and, wherever applicable, as a bridge between more specialised terminological resources.

For the concept layer, the latest *International Full* version of SNOMED CT will be used. For the language layer, SNOMED CT provides English labels but none of the other languages of the project are supported yet by SNOMED. Therefore, labels in these languages will either not be provided or only if translated within the project on a case-by-case basis (the translation of the hundreds of thousands of SNOMED CT terms is obviously out of the scope of the project).

6.4. LOINC Support

The project will support LOINC version 2.66 (https://loinc.org). The entire LOINC of around 94.000 concepts will be formalised and imported in a way similar to ICD, with "loinc:" prefixes used in front of labels representing codes. As from https://loinc.org/international/, translations of LOINC labels seem to be available in Italian, Belgian French, Greek, but not Romanian. Therefore, the possibility of having Romanian translations for LOINC needs to be investigated.

6.5. ATC Support

The *WHO Anatomical Therapeutic Chemical Classification System* (https://www.whocc.no/) is an international coding system for pharmaceutical substances. ATC is widely used for interoperability through the mapping of national coding systems (such as AIC for Italy or BNF for the UK). Contrary to other coded values, ATC codes will be encoded as *entities*: this allows the representation of rich structured knowledge about substances beyond their code and their name, such as the *defined daily dose*.

6.6. General Language Support

As EHRs often contain words and expressions from general language that are not covered by domain-specific terminologies, health knowledge is extended with the words and word senses from the general vocabulary. For this *wordnets* will be used for all five supported languages. Wordnets define the most widely used nouns, verbs, adjectives, and adverbs from the general vocabulary. Our source of multilingual wordnets is *Open Multilingual Wordnet* (http://compling.hss.ntu.edu.sg/omw/).

7. CONCLUSIONS AND NEXT STEPS

This document provided the first (v1) specifications for the IEHR Data Conversion Services. The specifications will continue to evolve as development progresses and the precise project needs are discovered, especially relating to the data conversion needs driven by the three scenarios.

In particular, an active collaboration (and sometimes co-development) with project partners will be necessary in the following areas:

- with hospitals: formalise and upload multilingual medical knowledge---both local and international---in four languages, such as the translation of FHIR attributes (IT, RO, EL, FR);
- with SIVECO: provide requirements with respect to semantic interoperability for the design of the HCP App GUI and back-end.

REFERENCES

- **[D2.4]** InteropEHRate Consortium, InteropEHRate Architecture v1 (deliverable D2.4). https://www.interopehrate.eu/resources/
- **[D2.7]** InteropEHRate Consortium, Interoperability Profile v1 (deliverable D2.7). https://www.interopehrate.eu/resources/
- **[D5.7]** InteropEHRate Consortium, Design of the Data Integration Platform v1 (deliverable D5.7). https://www.interopehrate.eu/resources/
- **[D5.11]** InteropEHRate Consortium, Design of information extractor and natural language translator v1 (deliverable D5.11). https://www.interopehrate.eu/resources/